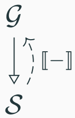
For the Metatheory of Type Theory, Internal Sconing is Enough

Rafaël Bocquet, Ambrus Kaposi, Christian Sattler

April 24, 2023
EuroProofNet WG6 meeting, Vienna

Motivation

Goal: Prove properties of the syntax of type theory from its universal property.
Syntax $\mathcal{S}$ = initial model of type theory.

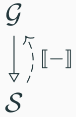


Problem 1: Unfolded definition of $\mathcal{G}$ is difficult / not modular.
$\rightsquigarrow$ Construct $\mathcal{G} \twoheadrightarrow \mathcal{S}$ by categorical gluing.

Problem 2: The section $[\![-]\!]$ is not always enough.
e.g. For normalization: need to combine with interpretation of variables.

Motivation

Goal: Prove properties of the syntax of type theory from its universal property.
Syntax $\mathcal{S}$ = initial model of type theory.

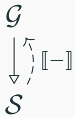


Problem 1: Unfolded definition of $\mathcal{G}$ is difficult / not modular.
$\rightsquigarrow$ Construct $\mathcal{G} \twoheadrightarrow \mathcal{S}$ by categorical gluing.

Problem 2: The section $[\![-]\!]$ is not always enough.
e.g. For normalization: need to combine with interpretation of variables.

Motivation

Goal: Prove properties of the syntax of type theory from its universal property.
Syntax $\mathcal{S}$ = initial model of type theory.

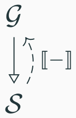


Problem 1: Unfolded definition of $\mathcal{G}$ is difficult / not modular.
$\rightsquigarrow$ Construct $\mathcal{G} \twoheadrightarrow \mathcal{S}$ by categorical gluing.

Problem 2: The section $[\![-]\!]$ is not always enough.
e.g. For normalization: need to combine with interpretation of variables.

Motivation

Goal: Prove properties of the syntax of type theory from its universal property.
Syntax $\mathcal{S}$ = initial model of type theory.



Problem 1: Unfolded definition of $\mathcal{G}$ is difficult / not modular.
$\rightsquigarrow$ Construct $\mathcal{G} \twoheadrightarrow \mathcal{S}$ by categorical gluing.

Problem 2: The section $[\![-]\!]$ is not always enough.
e.g. For normalization: need to combine with interpretation of variables.

Normalization

Running example: Normalization for small dependent type theory $\mathcal{T}$.

Syntax $\mathcal{S}$ = Initial object of $\mathbf{Mod}_{\mathcal{T}}$.

Canonicity proof:
universal property of $\mathcal{S}$ + sconing.

Normalization proof:
universal property of $\mathcal{S}$ + gluing + interpretation of variables.

Key idea: use universal property of other model $\mathcal{S}_F$ + sconing over $\mathcal{S}_F$.

Normalization

Running example: Normalization for small dependent type theory $\mathcal{T}$.

Syntax $\mathcal{S}$ = Initial object of $\mathbf{Mod}_{\mathcal{T}}$.

Canonicity proof:
universal property of $\mathcal{S}$ + sconing.

Normalization proof:
universal property of $\mathcal{S}$ + gluing + interpretation of variables.

Key idea: use universal property of other model $\mathcal{S}_F$ + sconing over $\mathcal{S}_F$.

Normalization

Running example: Normalization for small dependent type theory $\mathcal{T}$.

Syntax $\mathcal{S}$ = Initial object of $\mathbf{Mod}_{\mathcal{T}}$.

Canonicity proof:
universal property of $\mathcal{S}$ + sconing.

Normalization proof:
universal property of $\mathcal{S}$ + gluing + interpretation of variables.

Key idea: use universal property of other model $\mathcal{S}_F$ + sconing over $\mathcal{S}_F$.

Presheaf categories

Functor from renamings into syntax:

$$F : \mathbf{Ren}_{\mathcal{S}} \to \mathcal{S}$$

- The syntax/initial model $\mathcal{S}$ is an external object.
- Types and terms are stable under substitutions.
  $\rightsquigarrow$ Live in $\mathbf{Psh}(\mathcal{S})$.
- Normal forms are stable under renamings.
  $\rightsquigarrow$ Live in $\mathbf{Psh}(\mathbf{Ren}_{\mathcal{S}})$.

(Motives + Methods) for normalization also live in $\mathbf{Psh}(\mathbf{Ren}_{\mathcal{S}})$.

Normalization function should also be in $\mathbf{Psh}(\mathbf{Ren}_{\mathcal{S}})$.

Key idea

$$F : \mathbf{Ren}_{\mathcal{S}} \to \mathcal{S}$$

Define: Internal model $\mathcal{S}_F$ in $\mathbf{Psh}(\mathbf{Ren}_{\mathcal{S}})$ such that:

(Closed terms of $\mathcal{S}_F$) $\approx$ (Open terms of $\mathcal{S}$ over the image of F)

Normalization for closed terms of $\mathcal{S}_F$:
Every closed term $(1 \vdash a : A) \in \mathcal{S}_F$ admits a unique normal form.

Normalization proof: universal property of $\mathcal{S}_F$ + sconing over $\mathcal{S}_F$.

Universal property of $\mathcal{S}_F$ / Relative Induction Principle

1. Universal property of $\mathcal{S}_F$ / Relative Induction Principle

2. (Internal) Sconing

3. Construction of $\mathcal{S}_F$

(First-order) Models

Models of $\mathcal{T}$ = CwFs + type-theoretic structures (Π-types, etc.).

Models of a first-order essentially algebraic theory $\mathcal{T}^{\mathsf{fo}}$.
(Classified by a finitely complete category.)

Syntax $\mathcal{S}$ = Initial object of $\mathbf{Mod}_{\mathcal{T}}$.

Renamings

Definition

A **renaming algebra** over $\mathcal{S}$ is a pair $(\mathcal{R}, F)$, where $F : \mathcal{R} \to \mathcal{S}$ is a CwF morphism that is bijective on types.

Definition

The category of renamings $\mathbf{Ren}_{\mathcal{S}}$ is the initial renaming algebra over $\mathcal{S}$.

Internally to $\mathbf{Psh}(\mathbf{Ren}_{\mathcal{S}})$:

$$\mathsf{Var} : (1 \vdash A \text{ type}) \in \mathcal{S}_F \to \mathrm{Set}, \qquad \text{(Terms of the CwF } \mathbf{Ren}_{\mathcal{S}}\text{)}$$

$$\mathsf{var}_A : (x : \mathsf{Var}(A)) \to (1 \vdash \mathsf{var}(x) : A) \in \mathcal{S}_F. \qquad \text{(Action of } F \text{ on terms)}$$

$\mathsf{var}_A(x)$ is a closed term of $\mathcal{S}_F$!

Relative induction principle

Relative Induction Principle = Induction principle for $\mathcal{S}_F$.

Theorem

If $\mathcal{M}^\bullet$ *is a* **global** *displayed model over* $\mathcal{S}_F$ *and we additionally have*

$$\mathsf{var}^\bullet : \forall (A^\bullet : \mathcal{M}^\bullet.\mathsf{Ty}^\bullet(1^\bullet, A))\ (x : \mathsf{Var}(A)) \to \mathcal{M}^\bullet.\mathsf{Tm}^\bullet(1^\bullet, A^\bullet, \mathsf{var}_A(x)),$$

(Interpretation of variables in $\mathcal{M}^\bullet$)

then $\mathcal{M}^\bullet$ *admits a section* $[\![-]\!] : \mathcal{S}_F \to \mathcal{M}^\bullet$ *such that*

$$[\![\mathsf{var}(x)]\!] = \mathsf{var}^\bullet([\![A]\!], x).$$

Combines universal properties of $\mathcal{S}$ and $\mathbf{Ren}_{\mathcal{S}}$.

(Internal) Sconing

1. Universal property of $\mathcal{S}_F$ / Relative Induction Principle
2. (Internal) Sconing
3. Construction of $\mathcal{S}_F$

Higher-order models

A **higher-order model** of $\mathcal{T}$ is a family with additional structure:

$$\begin{aligned}
&\mathsf{Ty} : \mathrm{Set}, \\
&\mathsf{Tm} : \mathsf{Ty} \to \mathrm{Set}, \\
&\Pi : (A : \mathsf{Ty})\ (B : \mathsf{Tm}(A) \to \mathsf{Ty}) \to \mathsf{Ty}, \\
&\mathsf{app} : \mathsf{Tm}(\Pi(A, B)) \cong ((a : \mathsf{Tm}(A)) \to \mathsf{Tm}(B(a))).
\end{aligned}$$

Higher-order models support Higher-Order Abstract Syntax.

Higher-order models are models of a **higher-order theory** $\mathcal{T}^{\mathsf{ho}}$.
(Classified by a locally cartesian closed category.)

Higher-order and first-order models

Higher-order models:

- Defining higher-order models is "easy" (few components).
- There is no (suitable) category of higher-order models.

First-order models:

- Defining first-order models directly is "hard" (naturality conditions, . . .)
- The category $\mathbf{Mod}_{\mathcal{T}}$ of first-order models is well-behaved. It is complete, cocomplete, etc.

Construct first-order models from higher-order models.

Higher-order and first-order models

Higher-order models:

- Defining higher-order models is “easy” (few components).
- There is no (suitable) category of higher-order models.

First-order models:

- Defining first-order models directly is “hard”
  (naturality conditions, . . .)
- The category $\mathbf{Mod}_{\mathcal{T}}$ of first-order models is well-behaved.
  It is complete, cocomplete, etc.

Construct first-order models from higher-order models.

Standard model

First-order model **Set**.

- Underlying category is **Set**.
- Types are functions $\Gamma \to \mathrm{Set}$.
- Terms are dependent functions $(\gamma : \Gamma) \to A(\gamma)$.
- Context extensions are dependent sums: $\Gamma.A = (\gamma : \Gamma) \times A(\gamma)$.
- $\Pi(A, B) = \lambda\gamma \mapsto ((a : A(\gamma)) \to B(\gamma, a))$.

Set-contextualization

First-order model $\mathbf{Set}_{\mathbb{M}}$ for a higher-order model $\mathbb{M}$.

- Underlying category is **Set**.
- Types are functions $\Gamma \to \mathbb{M}.\mathsf{Ty}$.
- Terms are dependent functions $(\gamma : \Gamma) \to \mathbb{M}.\mathsf{Tm}(A(\gamma))$.
- Context extensions are dependent sums: $\Gamma.A = (\gamma : \Gamma) \times \mathbb{M}.\mathsf{Tm}(A(\gamma))$.
- $\Pi(A, B) = \lambda\gamma \mapsto \mathbb{M}.\Pi(A(\gamma), \lambda a \mapsto B(\gamma, a))$.

Contextualization adds contexts to a higher-order model.

Displayed higher-order model

Displayed higher-order model over **first-order model** $\mathcal{M}$.

(Motives and methods over closed terms of $\mathcal{M}$.)

$$
\begin{aligned}
\mathsf{Ty}^\bullet &: \mathcal{M}.\mathsf{Ty}(1) \to \mathrm{Set}, \\
\mathsf{Tm}^\bullet &: \mathsf{Ty}^\bullet(A) \to \mathcal{M}.\mathsf{Tm}(1, A) \to \mathrm{Set}, \\
\Pi^\bullet &: (A^\bullet : \mathsf{Ty}^\bullet(A)) \\
&\quad (B^\bullet : \forall (a : \mathcal{M}.\mathsf{Tm}(1, A))\ (a^\bullet : \mathsf{Tm}^\bullet(A^\bullet, a)) \to \mathsf{Ty}^\bullet(B[a])) \\
&\quad \to \mathsf{Ty}^\bullet(\mathcal{M}.\Pi(A, B)), \\
&\ldots
\end{aligned}
$$

Here $B : \mathcal{M}.\mathsf{Ty}(1.A)$ is dependent over A.

Scone-contextualization

$$\begin{array}{ccc} \mathbb{M}^{\bullet} & \xmapsto{\textbf{Scone}\text{-contextualization}} & \textbf{Scone}_{\mathbb{M}^{\bullet}} \\ \downarrow & & \downarrow \\ \mathcal{M} & & \mathcal{M} \end{array}$$

$\mathbb{M}^{\bullet}$ is a displayed higher-order model over $\mathcal{M}$.

$\textbf{Scone}_{\mathbb{M}^{\bullet}}$ is a displayed first-order model over $\mathcal{M}$.

Underlying category of $\textbf{Scone}_{\mathbb{M}^{\bullet}}$ is the **scone** of $\mathcal{M}$.

Set-contextualization = **Scone**-contextualization over $1_{\mathbf{Mod}_{\mathcal{T}}}$.

Canonicity proof structure

1. **Define:** Displayed higher-order model $\mathbb{S}^\bullet$ over $\mathcal{S}$.

$$\begin{aligned} \mathbb{S}^\bullet.\mathsf{Ty}^\bullet(A) &= \mathcal{S}.\mathsf{Tm}(1, A) \to \mathrm{Set}_0, \\ \mathbb{S}^\bullet.\mathsf{Tm}^\bullet(A^\bullet, a) &= A^\bullet(a), \\ \ldots \end{aligned}$$

2. **Construct: Scone**-contextualization $\mathbf{Scone}_{\mathbb{S}^\bullet} \twoheadrightarrow \mathcal{S}$.
3. **Obtain:** Section $[\![-]\!] : \mathcal{S} \to \mathbf{Scone}_{\mathbb{S}^\bullet}$.

If $(1 \vdash b : \mathsf{Bool}) \in \mathcal{S}$, then $[\![b]\!] : (b = \mathsf{true}) + (b = \mathsf{false})$.

Normalization proof structure

1. **Define:** Displayed higher-order model $\mathbb{S}^\bullet$ over $\mathcal{S}_F$.
   + Interpretation $\mathsf{var}^\bullet$ of variables.

$$\mathbb{S}^\bullet.\mathsf{Ty}^\bullet(A) = \{\text{logical predicate over } A + \text{unquote} + \text{quote}\}$$
$$\mathbb{S}^\bullet.\mathsf{Tm}^\bullet(A^\bullet, a) = A^\bullet_p(a)$$
$$\ldots$$
$$\mathsf{var}^\bullet(A^\bullet, x) = A^\bullet_u(\mathsf{var}^{\mathsf{ne}}(x)).$$

2. **Construct: Scone**-contextualization $\mathbf{Scone}_{\mathbb{S}^\bullet} \twoheadrightarrow \mathcal{S}$.
3. **Obtain:** Section $[\![-]\!] : \mathcal{S}_F \to \mathbf{Scone}_{\mathbb{S}^\bullet}$ s.t. $[\![\mathsf{var}_A(x)]\!] = \mathsf{var}^\bullet([\![A]\!], x)$.

If $(1 \vdash a : A) \in \mathcal{S}_F$, then $[\![A]\!]_q([\![a]\!]) : \mathsf{HasNf}(a)$.

$[\![-]\!]$ has computation rules.
$\rightsquigarrow$ Uniqueness of normal forms by induction on normal forms.

Normalization proof structure

1. **Define:** Displayed higher-order model $\mathbb{S}^\bullet$ over $\mathcal{S}_F$.
   $+$ Interpretation $\mathsf{var}^\bullet$ of variables.

$$\mathbb{S}^\bullet.\mathsf{Ty}^\bullet(A) = \{\text{logical predicate over } A + \text{unquote} + \text{quote}\}$$
$$\mathbb{S}^\bullet.\mathsf{Tm}^\bullet(A^\bullet, a) = A^\bullet_p(a)$$
$$\ldots$$
$$\mathsf{var}^\bullet(A^\bullet, x) = A^\bullet_u(\mathsf{var}^{\mathsf{ne}}(x)).$$

2. **Construct: Scone**-contextualization $\mathbf{Scone}_{\mathbb{S}^\bullet} \twoheadrightarrow \mathcal{S}$.
3. **Obtain:** Section $[\![-]\!] : \mathcal{S}_F \to \mathbf{Scone}_{\mathbb{S}^\bullet}$ s.t. $[\![\mathsf{var}_A(x)]\!] = \mathsf{var}^\bullet([\![A]\!], x)$.

If $(1 \vdash a : A) \in \mathcal{S}_F$, then $[\![A]\!]_q([\![a]\!]) : \mathsf{HasNf}(a)$.

$[\![-]\!]$ has computation rules.

$\rightsquigarrow$ Uniqueness of normal forms by induction on normal forms.

Normalization proof structure

1. **Define:** Displayed higher-order model $\mathbb{S}^\bullet$ over $\mathcal{S}_F$.
   + Interpretation $\mathsf{var}^\bullet$ of variables.

$$\mathbb{S}^\bullet.\mathsf{Ty}^\bullet(A) = \{\text{logical predicate over } A + \text{unquote} + \text{quote}\}$$
$$\mathbb{S}^\bullet.\mathsf{Tm}^\bullet(A^\bullet, a) = A^\bullet_p(a)$$
$$\ldots$$
$$\mathsf{var}^\bullet(A^\bullet, x) = A^\bullet_u(\mathsf{var}^{\mathsf{ne}}(x)).$$

2. **Construct: Scone**-contextualization $\mathbf{Scone}_{\mathbb{S}^\bullet} \twoheadrightarrow \mathcal{S}$.
3. **Obtain:** Section $[\![-]\!] : \mathcal{S}_F \to \mathbf{Scone}_{\mathbb{S}^\bullet}$ s.t. $[\![\mathsf{var}_A(x)]\!] = \mathsf{var}^\bullet([\![A]\!], x)$.

If $(1 \vdash a : A) \in \mathcal{S}_F$, then $[\![A]\!]_q([\![a]\!]) : \mathsf{HasNf}(a)$.

$[\![-]\!]$ has computation rules.
$\rightsquigarrow$ Uniqueness of normal forms by induction on normal forms.

Construction of $\mathcal{S}_F$

1. Universal property of $\mathcal{S}_F$ / Relative Induction Principle
2. (Internal) Sconing
3. Construction of $\mathcal{S}_F$

Models of first-order theories in presheaf categories

The following are equivalent:

- Models of $\mathcal{T}$ internally to $\mathbf{Psh}(\mathcal{C})$.
- Functors $\mathcal{C}^{\mathsf{op}} \to \mathbf{Mod}_{\mathcal{T}}$.

Interface between internal and external objects.

Holds for any first-order essentially algebraic theory.

Models of first-order theories in presheaf categories

The following are equivalent:

- Models of $\mathcal{T}$ internally to $\mathbf{Psh}(\mathcal{C})$.
- Functors $\mathcal{C}^{\mathsf{op}} \to \mathbf{Mod}_{\mathcal{T}}$.

Interface between internal and external objects.

Holds for any first-order essentially algebraic theory.

Construction 1 of $\mathcal{S}_F$

$$\mathcal{S} \xmapsto{\text{internalization}} \mathbb{S} \xmapsto[\text{contextualization}]{\text{telescopic}} \mathbf{Tele}_{\mathbb{S}} \xmapsto{\text{restriction}} F^*(\mathbf{Tele}_{\mathbb{S}})$$

- $\mathcal{S}$ is an external FOM.
- $\mathbb{S}$ is an internal HOM in $\mathbf{Psh}(\mathcal{S})$.
- $\mathbf{Tele}_{\mathbb{S}}$ is an internal FOM in $\mathbf{Psh}(\mathcal{S})$.
- $F^*(\mathbf{Tele}_{\mathbb{S}})$ is an internal FOM in $\mathbf{Psh}(\mathbf{Ren}_{\mathcal{S}})$.

$$\mathcal{S}_F = F^*(\mathbf{Tele}_{\mathbb{S}}).$$

Construction 2 of $\mathcal{S}_F$

$$\mathcal{S}_F : \mathbf{Ren}_{\mathcal{S}}^{\mathsf{op}} \to \mathbf{Mod}_{\mathcal{T}},$$
$$\mathcal{S}_F(\Gamma) = (\mathcal{S} \sslash F(\Gamma))$$

(Contextual slice of $\mathcal{S}$ over $F(\Gamma)$)

Conclusion

- For any functor $F : \mathcal{C} \to \mathcal{S}$, internal model $\mathcal{S}_F$ in $\mathbf{Psh}(\mathcal{C})$ s.t.

  $$\text{(Closed terms of } \mathcal{S}_F) \approx \text{(Open terms of } \mathcal{S} \text{ over the image of } F).$$

- Universal property of $\mathcal{S}_F$ combines universal properties of $\mathcal{S}$ and $\mathcal{C}$.
- Properties of closed terms of $\mathcal{S}_F$ can be proven by sconing.
- Section $[\![-]\!] : \mathcal{S}_F \to \mathbf{Scone}_{\mathbb{S}^\bullet}$ has computation rules.

`https://arxiv.org/abs/2302.05190`

(empty)

Universal property of $\mathcal{S}_F$

$$\mathcal{S}_F : \mathbf{Ren}_{\mathcal{S}}^{\mathrm{op}} \to \mathbf{Mod}_{\mathcal{T}},$$
$$\mathcal{S}_F(\Gamma) = (\mathcal{S} /\!\!/ F(\Gamma)).$$

For every $\Gamma \in \mathbf{Ren}_{\mathcal{S}}$, $\mathcal{S}_F(\Gamma) \cong \mathcal{S}[\underline{\gamma} : \Gamma]$.

Universal property of $\mathcal{S}_F$ glues universal properties of all $\mathcal{S}_F(\Gamma)$.

Category of sections

$$
\begin{array}{ccc}
\mathbf{Sect}^{op}_{\mathcal{T}}[\mathcal{M}^\bullet] & \xrightarrow{[\![-]\!]_0} & \mathbf{Sect}^{op}_{\mathcal{T}} \\
\downarrow{\scriptstyle \pi_0} & \lrcorner & \downarrow \\
\mathbf{Ren}_{\mathcal{S}} & \xrightarrow{\mathcal{M}^\bullet} & \mathbf{DispMod}^{op}_{\mathcal{T}} \\
 & \searrow{\scriptstyle \mathcal{S}_F} & \downarrow \\
 & & \mathbf{Mod}^{op}_{\mathcal{T}}.
\end{array}
$$

To construct $[\![-]\!] : \mathbf{Ren}_{\mathcal{S}} \to \mathbf{Sect}^{op}_{\mathcal{T}}$:

Equip $\mathbf{Sect}^{op}_{\mathcal{T}}[\mathcal{M}^\bullet]$ with renaming algebra structure.

About Synthetic Tait Computability

$$\begin{array}{ccc} \mathbf{Ren}_{\mathcal{S}} & \longrightarrow & \mathbf{Ren}_{\mathcal{S}}[F] \\ & \searrow^{F} & \downarrow{\scriptstyle \widetilde{F}} \\ & & \mathcal{S} \end{array}$$

$\mathbf{Ren}_{\mathcal{S}}[F] =$ lax colimit of $F : \mathbf{Ren}_{\mathcal{S}} \to \mathcal{S}$.

$\mathbf{Psh}(\mathbf{Ren}_{\mathcal{S}}[F]) =$ Artin gluing of $F^* : \mathbf{Psh}(\mathcal{S}) \to \mathbf{Psh}(\mathbf{Ren}_{\mathcal{S}})$.

Work with $S_{\widetilde{F}}$ instead of S_F.

Example of computation with section over $\mathcal{S}_F$

$$
\begin{aligned}
&[\![\Pi(A,B)]\!]_q([\![\mathsf{lam}(b)]\!]) \\
&\quad = \Pi^\bullet_q([\![A]\!], \lambda a^\bullet \mapsto [\![B(\underline{a})]\!][\underline{a} \mapsto a^\bullet])(\lambda a^\bullet \mapsto [\![b(\underline{a})]\!][\underline{a} \mapsto a^\bullet]) \\
&\qquad\qquad \text{(by the computation rules for } [\![\Pi(-)]\!] \text{ and } [\![\mathsf{lam}]\!]) \\
&\quad = \mathsf{lam}^{\mathsf{nf}}(\lambda a \mapsto \mathsf{let}\ a^\bullet = [\![A]\!]_u(\mathsf{var}^{\mathsf{ne}}(a))\ \mathsf{in}\ ([\![B(\underline{a})]\!][\underline{a} \mapsto a^\bullet])_q([\![b(\underline{a})]\!][\underline{a} \mapsto a^\bullet]) \\
&\qquad\qquad \text{(by definition of } \Pi^\bullet_q) \\
&\quad = \mathsf{lam}^{\mathsf{nf}}(\lambda a \mapsto ([\![B(\underline{a})]\!][\underline{a} \mapsto [\![\mathsf{var}(a)]\!]])_q([\![b(\underline{a})]\!][\underline{a} \mapsto [\![\mathsf{var}(a)]\!]])) \\
&\qquad\qquad \text{(by the computation rule for } [\![\mathsf{var}(a)]\!]) \\
&\quad = \mathsf{lam}^{\mathsf{nf}}(\lambda a \mapsto [\![B[\mathsf{var}(a)]]\!]_q([\![b[\mathsf{var}(a)]]\!])) \qquad \text{(by the naturality of } [\![-]\!]) \\
&\quad = \mathsf{lam}^{\mathsf{nf}}(b^{\mathsf{nf}}). \qquad \text{(by the induction hypothesis for } b^{\mathsf{nf}})
\end{aligned}
$$